

# MowNiT Laboratorium 1

Michał Szafarczyk

## Treść zadania:

13. Przybliżoną wartość pochodnej funkcji  $f(x)$  w punkcie  $x$  można obliczyć ze wzoru:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

Wykorzystać ten wzór do obliczenia pochodnej funkcji  $f(x) = \sin(x) + \cos(3x)$  w punkcie  $x = 1$  dla  $h = 2^{-n}$  ( $n = 0, 1, 2, \dots, 40$ ). Wykonać obliczenia dla różnej precyzji zmiennych. Zwrócić uwagę na typ argumentów i wyników dla funkcji bibliotecznych wykorzystywanych w obliczeniach. Jak wytłumaczyć, że od pewnego momentu zmniejszenie wartości  $h$  nie poprawia przybliżenia wartości pochodnej? Jak zachowują się wartości  $1+h$ ? Obliczone przybliżenia pochodnej porównać z dokładną wartością pochodnej.

Środowiskiem, w którym będziemy pracować będzie Python 3.9. Dla testów użyjemy biblioteki numpy.

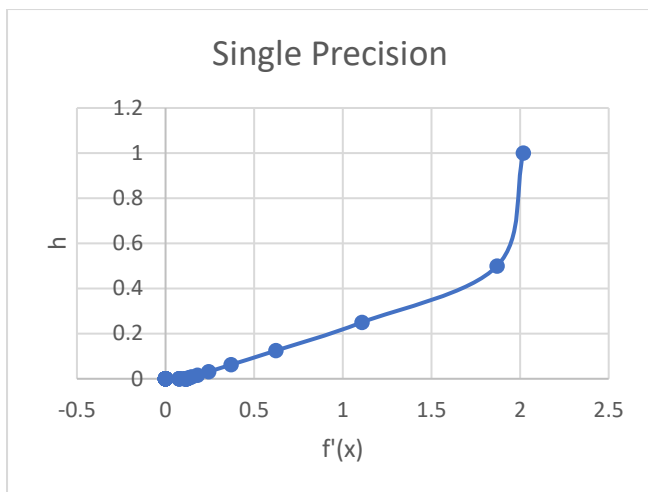
Za typy danych do testów posłużą nam np.single oraz np.double, które reprezentują liczby o odpowiednio pojedynczej i podwójnej precyzji. Funkcja  $f(x)$  została zrealizowana za pomocą np.sin oraz np.cos.

Single Precision		Double Precision		Single Precision			Double Precision		
h	f'(x)	h	f'(x)	n	h	1+h	n	h	1+h
1	2.017989174	1	2.017989225	0	1	2	0	1	2
0.5	1.870441353	0.5	1.870441398	1	0.5	1.5	1	0.5	1.5
0.25	1.107786983	0.25	1.107787095	2	0.25	1.25	2	0.25	1.25
0.125	0.623240876	0.125	0.623241279	3	0.125	1.125	3	0.125	1.125
0.0625	0.370399199	0.0625	0.370400066	4	0.0625	1.0625	4	0.0625	1.0625
0.03125	0.243442803	0.03125	0.243443074	5	0.03125	1.03125	5	0.03125	1.03125
0.015625	0.180097765	0.015625	0.180097563	6	0.015625	1.015625	6	0.015625	1.015625
0.0078125	0.148491758	0.0078125	0.148491395	7	0.007813	1.0078125	7	0.007813	1.0078125
0.00390625	0.132702191	0.00390625	0.132709114	8	0.003906	1.00390625	8	0.003906	1.00390625
0.001953125	0.124802708	0.001953125	0.124823693	9	0.001953	1.001953125	9	0.001953	1.001953125
0.000976563	0.120847088	0.000976563	0.120882477	10	0.000977	1.000976563	10	0.000977	1.000976563
0.000488281	0.118854466	0.000488281	0.11891225	11	0.000488	1.000488281	11	0.000488	1.000488281
0.000244141	0.117766714	0.000244141	0.117927234	12	0.000244	1.000244141	12	0.000244	1.000244141
0.00012207	0.117222867	0.00012207	0.11743475	13	0.000122	1.00012207	13	0.000122	1.00012207
6.10E-05	0.11695095	6.10E-05	0.117188514	14	6.10E-05	1.000061035	14	6.10E-05	1.000061035
3.05E-05	0.115838431	3.05E-05	0.117065397	15	3.05E-05	1.000030518	15	3.05E-05	1.000030518

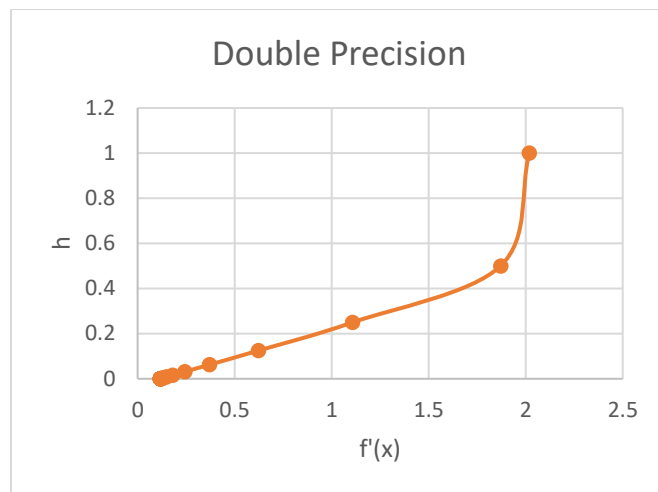
1.53E-05	0.115770453	1.53E-05	0.117003839	16	1.53E-05	1.000015259	16	1.53E-05	1.000015259
7.63E-06	0.115736465	7.63E-06	0.11697306	17	7.63E-06	1.000007629	17	7.63E-06	1.000007629
3.81E-06	0.10790697	3.81E-06	0.116957671	18	3.81E-06	1.000003815	18	3.81E-06	1.000003815
1.91E-06	0.107898473	1.91E-06	0.116949976	19	1.91E-06	1.000001907	19	1.91E-06	1.000001907
9.54E-07	0.076644224	9.54E-07	0.116946129	20	9.54E-07	1.000000954	20	9.54E-07	1.000000954
4.77E-07	0.0766421	4.77E-07	0.116944205	21	4.77E-07	1.000000477	21	4.77E-07	1.000000477
2.38E-07	0.076641038	2.38E-07	0.116943243	22	2.38E-07	1.000000238	22	2.38E-07	1.000000238
1.19E-07	0.076640506	1.19E-07	0.116942762	23	1.19E-07	1.000000119	23	1.19E-07	1.000000119
5.96E-08	0	5.96E-08	0.116942521	24	5.96E-08	1.00000006	24	5.96E-08	1.00000006
2.98E-08	0	2.98E-08	0.116942398	25	2.98E-08	1.00000003	25	2.98E-08	1.00000003
1.49E-08	0	1.49E-08	0.116942339	26	1.49E-08	1.000000015	26	1.49E-08	1.000000015
7.45E-09	0	7.45E-09	0.116942316	27	7.45E-09	1.000000007	27	7.45E-09	1.000000007
3.73E-09	0	3.73E-09	0.116942286	28	3.73E-09	1.000000004	28	3.73E-09	1.000000004
1.86E-09	0	1.86E-09	0.116942227	29	1.86E-09	1.000000002	29	1.86E-09	1.000000002
9.31E-10	0	9.31E-10	0.116942167	30	9.31E-10	1.000000001	30	9.31E-10	1.000000001
4.66E-10	0	4.66E-10	0.116942167	31	4.66E-10	1	31	4.66E-10	1
2.33E-10	0	2.33E-10	0.116941929	32	2.33E-10	1	32	2.33E-10	1
1.16E-10	0	1.16E-10	0.116941452	33	1.16E-10	1	33	1.16E-10	1
5.82E-11	0	5.82E-11	0.116941452	34	5.82E-11	1	34	5.82E-11	1
2.91E-11	0	2.91E-11	0.116939545	35	2.91E-11	1	35	2.91E-11	1
1.46E-11	0	1.46E-11	0.116943359	36	1.46E-11	1	36	1.46E-11	1
7.28E-12	0	7.28E-12	0.116928101	37	7.28E-12	1	37	7.28E-12	1
3.64E-12	0	3.64E-12	0.116943359	38	3.64E-12	1	38	3.64E-12	1
1.82E-12	0	1.82E-12	0.116882324	39	1.82E-12	1	39	1.82E-12	1
9.09E-13	0	9.09E-13	0.116821289	40	9.09E-13	1	40	9.09E-13	1

Tabela 1. – Wyniki pomiarów

**Dokładna wartość  $f'(1) = 0.11694228168853815$**



Wykres 1. – dla funkcji  $f'(x)$  (single)



Wykres 2. – dla funkcji  $f'(x)$  (double)

Rozbieżności w przypadku pojedynczej precyzji, dla dużych wartości  $n$  wynikają oczywiście z charakterystyki danej. Dla bardzo małych wartości  $h$ , pomimo, że program potrafi przechować samą wartość, to podczas wykonywania obliczeń błędy obliczeniowe sprawiają, że wynikiem jest 0.0. Numpy'owe single oraz double mają odpowiednio 8 bitów dla cechy, 23 bitów dla mantysy oraz 11 bitów dla cechy i 52 bity dla mantysy.

Funkcje  $\sin()$  oraz  $\cos()$  w bibliotece numpy zostały wykonane za pomocą przybliżenia wielomianem. (Ogólnie wielomianem, natomiast tutaj pokazałem akurat przybliżenie za pomocą szeregu Maclaurina, który sprawdziłby się przy obliczaniu wartości dla  $x$  blisko 0):

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$$

Jak łatwo zauważyć, dla bardzo małych przekazywanych wartości  $x$ , przez podnoszenie ich do kolejnych potęg, szybko rośnie nam ilość miejsc po przecinku. W związku z tym, ponieważ zarówno typ pojedynczej, jak i podwójnej precyzji, mogą pamiętać skończoną ilość miejsc po przecinku, pewna część liczby jest tracona, a całość zaokrąglana. W związku z tym, dla bardzo małego  $h$  program wylicza  $f(x+h)$  jako równe  $f(x)$  i w efekcie otrzymujemy wartość  $f'(x)$  jako równą 0.

Możemy również zauważyć co się dzieje w przypadku tych samych wartości  $h$ , jednak podczas prostego działania dodawania. Zaczynając od wartości rzędu  $10^{-10}$  działanie  $1+h$  zwraca 1, ignorując zupełnie wartość  $h$ .